Programación Python para Machine Learning

Lección 4: Selección de características

eip ESCUELA INTERNACIONAL POSGRADOS

ÍNDICE

Lección 4: Selección de características

1. Selección de características

Los conjuntos de datos pertenecientes a los problemas del mundo real son de una muy diversa naturaleza. Atendiendo a su tamaño, a veces son pequeños y manejables, pero otras veces son tremendamente grandes. Como cabría esperar, resulta muy dificultoso procesar los conjuntos de datos de gran tamaño. Se pueden considerar dos aspectos que, forzosamente, revierten en el tamaño de un conjunto de datos: las instancias y las características. Cuanto mayor sea el número de características, más grandes será el conjunto de datos. Lo mismo ocurre con las instancias, aunque haya conjuntos de datos en los que el número de características es muy elevado, pero que no contienen tantas instancias y al revés. No obstante, ante conjuntos de datos de tamaño considerable hay que plantearse cómo procesarlos del mejor modo posible.

Objetivos

- Entender el concepto de selección de características y las razones para su aplicación.
- Conocer los distintos tipos de técnicas para selección de características.
- Aprender a aplicar métodos *Filter* para selección de características.
- Aprender a aplicar métodos *Wrapper* para selección de características.
- Introducir los conceptos del PCA y su aplicación.

En un conjunto de datos de alta dimensión, lo más probable es que algunas características sean totalmente irrelevantes, insignificantes y sin importancia. La contribución de este tipo de características en el modelado predictivo es menor en comparación con las características críticas. También pueden tener una contribución nula. Incluso, en algunos casos, la inclusión de este tipo de atributos es contraproducente en la predicción. Por lo tanto, estas características causan una serie de problemas que impiden que el proceso de modelado predictivo se haga de modo eficiente.

En resumen, los problemas que genera el exceso de características tienen que ver con la necesidad extra de recursos, con el incremento tiempo a la hora del proceso de entrenamiento y con el ruido que introducen que repercute negativamente en el rendimiento del modelo de Machine Learning.

La **selección de características** es el proceso de selección de las características más significativas o relevantes de un conjunto de datos determinado. Las ventajas principales de la selección de características son:

- Se reduce la complejidad de un modelo y facilita su interpretación.
- Permite que el proceso de entrenamiento del modelo de Machine Learning sea más rápido.
- Mejora el rendimiento de los modelos.
- Reduce el sobreentrenamiento *(overfitting)*.

Existen tres perspectivas, o naturaleza de métodos, para llevar a cabo un proceso de selección de características:

1. Métodos «*Filter*».
2. Métodos «*Wrapper*».
3. Métodos embebidos.

2. MÉTODOS *FILTER* PARA SELECCIÓN DE CARACTERÍSTICAS

Los métodos *Filter* para la selección de características son grupo de técnicas que tratan la elección de un subconjunto de atributos sin incluir ningún algoritmo de Machine Learning, solo basándose en las propiedades estadísticas de los datos. Se utilizan generalmente como un paso de preprocesamiento. Las características se clasifican utilizando estadísticos que tienden a determinar la correlación de las características con la variable de objetivo. La correlación es un término altamente ligado al contexto, y varía de un trabajo a otro dependiendo de si el tipo de los datos son continuos o categóricos.

El módulo scikit-learn proporciona la clase `SelectKBest` que se puede configurar con diferentes test estadísticos para seleccionar un número específico de características. Dependiendo del tipo de variable, se debe configurar el método según estas cuatro posibilidades:

- Variables de entrada numéricas
  - Variable a predecir numérica (regresión)
    - Coeficiente de correlación de Pearson. Asume variables de que siguen una distribución normal. Correlación lineal.
    - Coeficiente de ranking de Spearman. Correlación no lineal.
  - Variable a predecir categórica (clasificación)
    - Coeficiente de correlación ANOVA. Lineal.
    - Coeficiente de ranking de Kendall. No lineal.
- Variables de entrada categóricas
  - Variable a predecir numérica (regresión)
    - Coeficiente de correlación ANOVA. Lineal.
    - Coeficiente de ranking de Kendall. No lineal.
  - Variable a predecir categórica (clasificación)
    - Test Chi-cuadrado.
    - Información mutua.

Estos test están implementados en scikit-learn:

- Coeficiente de correlación de Pearson: `f_regression()`
- ANOVA: `f_classif()`
- Chi-cuadrado: `chi2()`
- Información mutua: `mutual_info_classif()` y `mutual_info_regression()`

Además, en SciPy se pueden encontrar los estadísticos de Kendall y Spearman. Para determinar el test a utilizar dentro de `SelectKBest`, se debe asignar al parámetro `score_func` el test correspondiente.

Concretamente, en el siguiente código se observa cómo realizar el proceso en Python:

```
from pandas import read_csv
import pandas
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif


filename = 'Indian-Liver-Patient.csv'

col_names = ['age', 'gen', 'tbili', 'dbili', 'alkphos', 'sgpt',
'sgot', 'tp', 'alb', 'ag' ,'class']
data = read_csv(filename, names=col_names)

input_data = data[data.columns[:-1]]
input_data = pandas.get_dummies(input_data, prefix=['gen'])

filterKB = SelectKBest(score_func=f_classif, k=4)
filter_model = filterKB.fit(input_data, data['class'])


print(filter_model.scores_)
selected = filter_model.transform(input_data)
print(pandas.DataFrame(selected).head())
```

3. MÉTODOS *WRAPPER* PARA SELECCIÓN DE CARACTERÍSTICAS

Un método *Wrapper* de selección de características necesita de un algoritmo de Machine Learning, utilizando su rendimiento como criterio de evaluación del subconjunto de atributos seleccionados.

Este método pretende un acoplamiento idóneo entre el nuevo subconjunto y el método de Machine Learning. De modo iterativo, va tomando la decisión de añadir o eliminar variables en base al rendimiento de los sucesivos modelos entrenados.

El método más popular de este tipo es denominado por sus siglas en inglés RFE (Recursive Feature Elimination). Con una estrategia recursiva, va eliminando los atributos y entrenando un nuevo modelo con los atributos que restantes. En base al rendimiento del modelo, identifica qué atributos contribuyen más a predecir la variable objetivo.

```
from pandas import read_csv
import pandas
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression

filename = 'Indian-Liver-Patient.csv'

col_names = ['age', 'gen', 'tbili', 'dbili', 'alkphos', 'sgpt',
'sgot', 'tp', 'alb', 'ag' ,'class']
data = read_csv(filename, names=col_names)

data.fillna(0, inplace=True)

input_data = data[data.columns[:-1]]
input_data = pandas.get_dummies(input_data, prefix=['gen'])

logReg = LogisticRegression(solver='liblinear')
rfe = RFE(logReg)
rfe_model = rfe.fit(input_data, data['class'])
print("Número de características seleccionadas: ",
rfe_model.n_features_)
print("Atributos seleccionados: ", rfe_model.support_)
print("Ranking de características: ", rfe_model.ranking_)
```

4. MÉTODOS EMBEBIDOS PARA SELECCIÓN DE CARACTERÍSTICAS

Conceptualmente, los métodos embebidos realizan la selección de variables como parte del procedimiento de entrenamiento y normalmente están supeditados a un modelo de aprendizaje concreto. Algunos ejemplos de métodos embebidos son los árboles de clasificación o *Random Forests*.

5. Consideraciones prácticas

5.1. Comparativa de métodos de selección de características

Gráficamente, la Ilustración 1: Esquema general de los métodos filter y los métodos wrapper para selección de características. muestra las diferencias estructurales entre los esquemas de las técnicas *Filter* y *Wrapper*.

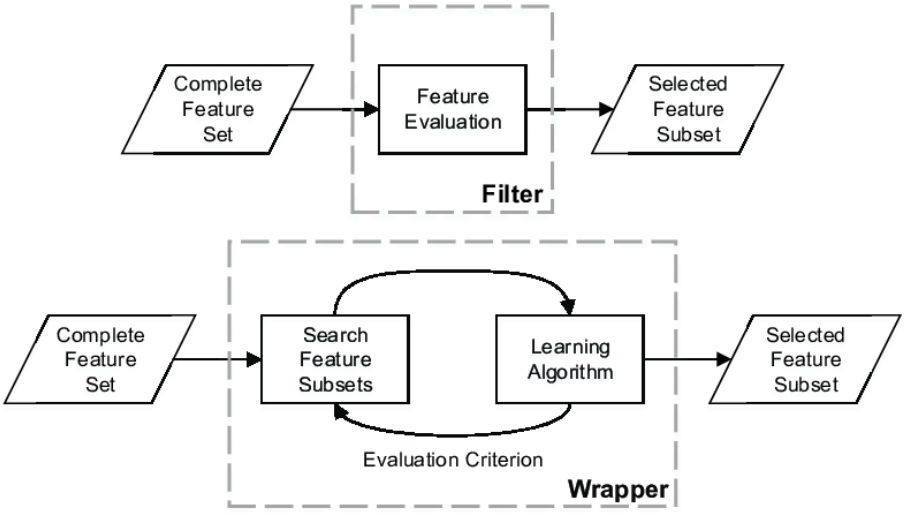

Ilustración 1: Esquema general de los métodos filter y los métodos wrapper para selección de características. [Yan Mei et al., 2015]

Los métodos *Filter* son mucho más rápidos que los *Wrapper*, ya que los primeros no necesitan entrenar ningún modelo. Por esta misma razón, los últimos son mucho más costosos computacionalmente.

Los métodos *Filter* utilizan métodos estadísticos para evaluar los subconjuntos, mientras que los *Wrapper* utilizan validación cruzada normalmente sobre el modelo que finalmente se va a acabar usando.

En muchas ocasiones, los métodos *Filter* pueden alejarse del mejor subconjunto de características, mientras que los *Wrapper*, debido a su acoplamiento al modelo predictivo, suelen obtener una combinación óptima.

> El temido *overfitting* es más probable que aparezca con modelos entrenados a partir de característica seleccionadas a partir de métodos *Wrapper*.

5.2. En qué punto aplicar la selección de características

Una vez vista la definición, tipos y uso de las técnicas de selección de características en un proyecto de Machine Learning, es muy importante entender en qué punto exacto debe integrarse este paso en el proceso general. La respuesta más directa es que se debería incluir justo antes de proporcionar los datos de entrenamiento al modelo. Esto se hace especialmente importante cuando se utilizan métodos de estimación de parámetros como la validación cruzada. En tal caso, hay que asegurarse que la selección de características se realiza sobre el *fold* de datos de entrenamiento. El error sería que la selección de características se realizara antes de particionar los datos en *train/test*. Si se realiza la selección de características con todos los datos y luego se hace una validación cruzada, entonces los datos de test en cada *fold* del procedimiento de validación también han sido utilizados para elegir las características, hecho que sesga el rendimiento del modelo.

6. Análisis de Componentes Principales (PCA)

Pese a que esta lección está destinada a los métodos de selección de características, merece la pena incluir una técnica íntimamente relacionada con ellos, aunque de naturaleza distinta. Es el denominado Análisis de Componentes Principales (PCA), que propiamente es una técnica de reducción de la dimensionalidad, pese a que en muchos lugares se le considera erróneamente como una selección de atributos. Se trata de una técnica utilizada para describir un conjunto de datos en términos de nuevas variables (componentes) no correlacionadas y ortogonales entre sí. Las componentes se ordenan por la cantidad de varianza original que pueden describir, por lo que la técnica es útil para reducir la dimensionalidad de un conjunto de datos. El PCA convierte un conjunto de variables posiblemente correlacionadas en un conjunto de variables sin correlación lineal llamadas "componentes principales".

Esta técnica de transformación lineal es muy utilizada para el análisis exploratorio de los datos. El módulo scikit-learn cuenta con la clase **PCA** para implementar la mayoría de las funcionalidades necesarias para crear y utilizar modelos PCA.

```
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.decomposition import PCA

iris = datasets.load_iris()
X = iris.data
y = iris.target
target_names = iris.target_names

pca = PCA(n_components=2)
X_r = pca.fit(X).transform(X)
print('Varianza explicada (primeras 2 componentes): ',
pca.explained_variance_ratio_)

plt.figure()
colors = ['indigo', 'lightseagreen', 'gold']
lw = 2
for color, i, target_name in zip(colors, [0, 1, 2], target_names):
    plt.scatter(X_r[y == i, 0], X_r[y == i, 1], color=color, alpha=.8,
lw=lw, label=target_name)
plt.legend(loc='best',  scatterpoints=1)
plt.title('PCA del dataset IRIS')
plt.show()
```

El resultado de aplicar un PCA sobre el dataset Iris es el que se puede ver en la Ilustración 2: PCA sobre el dataset Iris. La primera componente explica el 0.9246 de la varianza, mientras que la segunda componente explica el 0.0531.

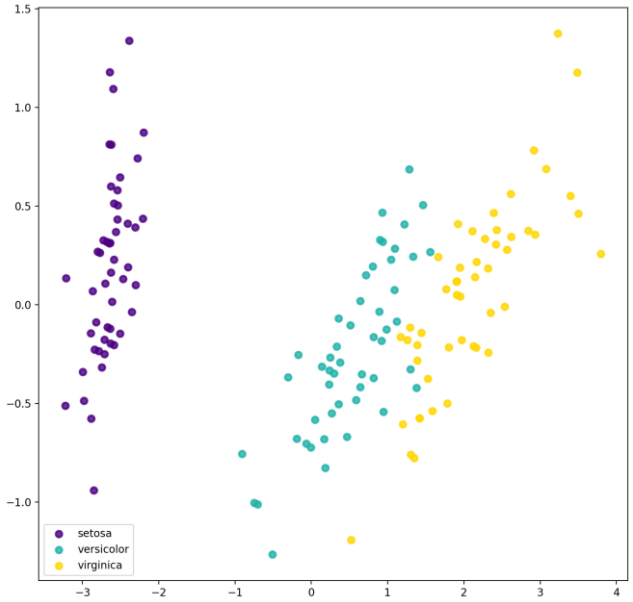

Ilustración 2: PCA sobre el dataset Iris.

7. Puntos Clave

En esta lección hemos aprendido:

- Comprender el concepto de selección de características y las razones para su aplicación.
- Profundizar en los principios de los distintos tipos de técnicas para selección de características.
- Aplicar métodos Filter para selección de características en un conjunto de datos mediante scikit-learn.
- Aplicar métodos Wrapper para selección de características en un conjunto de datos mediante scikit-learn.
- Introducir los conceptos del PCA y su aplicación en un conjunto de datos mediante scikit-learn.